Oscar Alvarez III

Dr. Emmet Tomai

CMPE 3300.01

9 May 2016

NASA Pathways Internship: Spring 2016

I was selected to contribute to the Data Systems and Handling Branch under the Avionics Flight Systems Division at the Lyndon B. Johnson Space Center in Houston, Texas. There I used my knowledge from school, as well as my job experience from the military, to help me comprehend my assigned project and contribute to it. With help from my mentors, supervisors, colleagues, and an excellent NASA work environment, I was able to learn, as well as accomplish, a lot towards my project. Not only did I understand more about embedded systems, microcontrollers, and low-level programming, I also was given the opportunity to explore the NASA community.

My specific role, as a software computer engineer, was to contribute to the integration of a resistive touch LCD screen into the Modular Integrated Stackable Layers (MISL) system used for current and future flight systems in space flight vehicles (International Space Station/ORION). The MISL is a stack of boards comprised of the MSP430 microcontroller, a power board, an Ethernet board, and a pinout board used to connect to a single, or multiple devices depending on the desired function (In my case, it was an LCD screen with a CO2/temp/humidity sensor).

I was assigned to understand embedded systems, the MSP430 microcontroller, and low-level programming, and apply that knowledge into using an LCD screen that displayed dynamic sensor data while storing that data for analysis. The idea was to be able to view sensor data, with a brief history, on the LCD screen. The touch screen would be used to change between different sensor readings, view in different graphing formats, and to scroll through data history. I didn't have to start from scratch, at least. A previous pathways intern, from a previous semester, developed most of the drivers needed to send information to the LCD screen. Other drivers used to initialize the MISL stack were developed by NASA computer scientists/engineers (I didn't have to worry about initializing clocks or pin assignments). Unfortunately, I was unable to contribute to the touch screen capabilities. The LCD backpack (A board used to communicate between the LCD and the MSP430) was only configured for non-touchscreen LCDs; data could only be sent to the LCD and not received from. An OSSI intern (Temporary interns) was assigned in designing a capable backpack and should be done by the end of this semester.

My first assignment was to understand the MSP430, Inter Integrated Circuit (I2C), and Serial Peripheral Interface (SPI). I had no hands on opportunities, with the hardware or software, since the only computer science guru was on annual leave for 2 weeks. I educated myself by looking up documentation and datasheets. Once I found out that I couldn't receive information from the LCD screen due to the current capabilities of the hardware, I stopped focusing on I2C/SPI. I then focused my attention on understanding the MSP430 and how to communicate with it. I used Code Composer Studio (CCS), a code editor developed by Texas Instruments made specific for MSP microcontroller families, and took the initiative by asking for a MISL stack and an LCD screen (which I put together myself) to explore with.

At the beginning of my tour, my project was not yet given to me and most of my first month was used exploring the code (source and header files made by the previous pathways intern and current NASA employees for the MSP430), which was done using C programming. I then grabbed a breadboard, a 1k resistor, some wires, and made a LED light blink. Once I felt confident enough, I decided to get ambitious. I recreated HAL9000 from the movie *2001: A Space Odyssey*. I displayed him on the LCD screen with the famous nameplate and his glowing eye, while famous quotes from the movie popped up every once in a while. I presented it to my supervisor and branch manager and explained to them that I now knew pixel manipulation, character display, screen orientation, RGB display, and a basic understanding of CCS. After that, my main project, described previously, was given to me.

I was asked to display a graph that would show dynamic sensor data on the LCD screen. Not having a sensor to work with, I used multiple random number generators in a single forever loop to simulate different sensor data inputs and displayed it on a bar graph. After I showed this to my supervisors, they then decided a line graph that showed a brief history would be more beneficial. That way the behavior of the sensor data could be monitored; any unusual or drastic changes, within a desired range, in the environment of a space vehicle would be visible via display and be immediately observed.

I understood right away that a dynamic line graph resembled a queue. The whole queue would be displayed and updated as new sensor data came in; truncating the old and inserting the new. The size of the queue was restricted since the width of the display was limited. I already had to display each item out every time the queue was updating so I didn't want to have to shift items in an array in addition to this (There was double display of the queue as well; I had to black out the old displayed queue and color in the new to update its position on the graph). The more obvious solution was to use a linked list and push the new while popping the old.

The linked list wouldn't compile. I went over the code and couldn't see where I went wrong. I was actually stumped on this for a while until I found out, from online forums, that I couldn't use a "class" linked list in C. Since I couldn't use a "class" linked list, I resorted to using a “struct” linked list. My code compiled, but when implemented it would only display 15 plots on my graph then start printing junk. Again, I was stumped on this. After not being able to find anything online on this matter I asked my mentor, the only CS personnel in our branch, for assistance. He taught me to navigate through the registers to see exactly where my information was being stored. What we found out was that we were running out of heap space. It would save up to 15 plots, and when it ran out of heap the pointer would point to some restricted registers that couldn’t be written over, but were still allowed to be pointed at (The junk that was being displayed was whatever was in those registers). CCS had a feature that allowed me to increase the heap space. Once this was accomplished, my dynamic line graph worked great.

Although, I had accomplished my line graph, I was further advised that using dynamic memory allocation on an embedded system, with limited memory, wasn’t the best method. I had to abandon my linked list idea and started looking into 2D arrays(X and Y coordinate elements). Again, I wanted to avoid shifting all the elements in the array and found that a circular array was the best solution; the newest plot was saved over the oldest. With the circular array implemented in my code, I was able to display my dynamic graph without a problem.

The next step was to actually send real data into the MSP430 and have my code display it to the LCD screen. I was given a csv file that contained real data, from in orbit flights, with CO2, temperature, and humidity readings. My supervisor wanted me to send this information from my computer. I used Visual Studio C++ to write a program that read the csv file and sent the data out the USB port(using the Serial Port Class) through the RS232 and then to the MSP430. For some reason, the MSP430 was not receiving the data via its receiving register. In order to find out if my code was actually being sent, I hooked up an oscilloscope to the RS232 and captured the first string of characters that were sent. I looked up the characters in the string on the ASCII table, converted their hex to binary, inverted the binary (RS232 inverts data), and read it backwards (little endian) to see if it matched the screen capture on the oscilloscope (which it did). This not only stumped me but my mentor as well. It turns out we just had to unplug the MISL stack and plug it back in again.

The second part of this step was to receive the data from the computer. As I was writing my code on CCS, I saw a problem. My code was being customized to read this data that was being sent from my computer, without knowing how the actual sensor sent the data. When I brought this up to my mentor and supervisor, they were able to get me the same kind of sensor used in space flight vehicles. I was able to receive the information from the sensor (CO2 filtered, CO2 unfiltered, temperature, humidity), use interrupts and buffers to read and distinguish the data into their corresponding variables (The sensor sent out one whole string, at a time, that included all four readings, which I had to translate and place into separate variables). I then converted the separate string variables into integers and used the methods provided in the data sheet to get the desired results (For example: x = 01221; x = (x – 1000)/10; // x = 22.1 degrees Celsius).

I have not sent this data into my line graph display yet, but have been able to create a digital display of the current temperature, humidity, and CO2 levels (both filtered and unfiltered). I still have one more week to accomplish this. Whatever isn't finished will get passed on to the next pathways intern. Looking at this doesn't seem like a lot, but I spent a lot of time on each step trying to learn, understand, implement, and solve every problem I encountered. I now have a better understanding and experience in microcontrollers, C programming, low-level coding, embedded systems, serial ports, interrupts, buffers, LCD Displays, sensors, oscilloscopes, soldering, etc. I believe this experience will contribute a lot to my growth as a Computer Engineer.

I am very grateful for the time I've been given here and have learned a lot from this experience. My entire tour wasn't just on my project alone. Besides the training courses, professional development courses, committee meetings, branch meetings, division meetings, lectures, and a fire drill, I was able to explore much more of NASA than I could ever hope for.

My tours (tours were during work hours and were heavily encouraged by management since, being pathways interns, we are able to select which location we wanted our next internship to be) included a trip to the Neutral Buoyancy Laboratory (NBL), Power and Propulsion Laboratory, Eagle Works Laboratory, Ellington Field, International Space Station Mock Ups, Chamber A with the James Webb Telescope (Hubble replacement) undergoing testing, Historic Mission Control, Mission Control, Future Mission Control, etc. I even got to attend a focus group of the greatest engineering minds at NASA working together to solve an out of this world problem (literally), watch them break it down, and work together to come up with solutions (I can't discuss anything specific).

Fortunately, as a pathways intern, my adventure does not stop. I will be transferred to the Flight Operations side of NASA to see what it's like to be trained as a flight controller, and given an opportunity to sit in Mission Control! I will start right after my spring tour is finished. I look forward to continuing my professional growth here at Johnson Space Center.